6K−4

Hyper−OSによるOSのデバッグ環境の研究

清水正明，並木美太郎，高橋延匡
(東京農工大学　工学部　電子情報工学科)

1．はじめに

OS などのシステム・プログラムは、信頼性が要求されることから、開発において、デバッグとテストに多大な労力を必要とする．特に OS は、ハードウェアの資源管理を行うことから、非同期処理に対するデバッグや再現性のあるテストを行うのが困難である．

従来の OS のデバッグは、デバッグ対象ごとにデバッグツールを用意したり、アドホックな手法で行なわれることが多い．これに対して、メインフレームでは、汎用大型計算機上で、VM370 に代表される仮想マシンを用いてデバッグを行う手法が開発された[1]．

近年、パーソナルコンピュータやワークステーションが普及し、これらの計算機上で、複数の OS が開発されることも珍しくない．このようなシステムのデバッグでは、OS デバッグのためのツールや手法を、可能な限り共有できることが望ましい．

本稿では、パーソナルコンピュータ、ワークステーションにおける、OS のデバッグ環境を構築することを目的とした、OS デバッガの開発について報告する．

2．OS のデバッグ環境の問題

OS というプログラムが応用プログラムと違う点は、ハードウェアの資源管理を行うことである．OS はハードウェアの非同期の動作により、再現性のない動きをする．したがって、OS のデバッグに要求されるのは、このハードウェアの管理機能をデバッグできる機能である．

OS のデバッグに要求される機能を次に示す．

(1)割込みのモニタ
(2)割込みの発生
(3)I/O のエミュレート
(4)OS の SVC のエミュレート
(5)メモリの監視
(6)再現性のない動作のロギングと、その再現
(7)TCB などの OS 内部の管理表の操作
(8)性能評価のためのパフォーマンスモニタ
(9)レジスタ操作、ブレークポイント、トレース、メモリダンプなど、従来のデバッガにある機能

上記の機能があれば、効果的な OS のデバッグが可能であると考えている．

我々の OS のデバッグ環境は、上記の機能を満たすことを目標にする．

3．OS デバッガの設計思想

当研究室では OS の研究を行っており、過去に CP/M-68K、OS/omicron 第 2 版、OMICRON V3 のデバッグを行ってきた．そのときの経験から、2 章で示した機能が必要であると考えている．

OS デバッガの設計目標を次に定めた．

(1)特定 OS のデバッグではなく、多種の OS のデバッグを行う環境を提供する

一つの OS デバッガで、多種の OS のデバッグを可能にすれば、新しい OS が、前の OS と同じ環境でデバッグを行うことができる．

(2)デバッグ対象 OS に特別の仕掛けを用意しない

OS に特別の仕掛けを用意しなくても、OS デバッガの仕掛けのみで、デバッグ可能である．

OS デバッガは、以上の設計目標で、2 章の機能を満たすように設計した．

4．OS デバッガの設計

設計目標を満足するために、OS デバッガを、デバッグ用ハイパ OS、デバッガ OS、デバッガという三層構成に分けた（図 1 ）．

ハイパ OS にすべての機能を閉じ込めて、ハイパ OS の中に OS デバッガを実現するというアプローチも選択できたが、次の理由により上記のような三層構成に分けた．

(1)デバッグする OS ごとに、OS デバッガを用意するのは大変である．OS デバッガの機能には、OS の種類に依存する機能と、OS の種類に依存しない機能があるはずである．例えば、TCB のサーチやダンプ、管理表のプリントアウト等の機能は、特定の OS だけの機能ではない．これらの機能は、共有したい．

したがって、OS デバッガの機能のうち、ある OS に依存する機能をデバッガに、それ以外の共通機能を OS 層に入れることにした．

(2)OS 層では、デバッガ用の入出力を提供する．しかし、この構成では、デバッグ対象 OS に細工をしないでデバッグをするのは難しい．

以上の二つの理由により、ハイパ OS とデバッガ OS では、ハイパ OS 上の OS をデバッグするためのプリミティブを用意して、そのプリミティブを使って、デバッガが、ある OS 専用のデバッグ機能を提供するという形をとることにした．

A Debugging Environment for Operating Systems on a Hyper−OS
Masaaki SHIMIZU, Mitarou NAMIKI and Nobumasa TAKAHASHI
Department of Computer Science, Faculty of Technology, Tokyo University of Agriculture and Technology

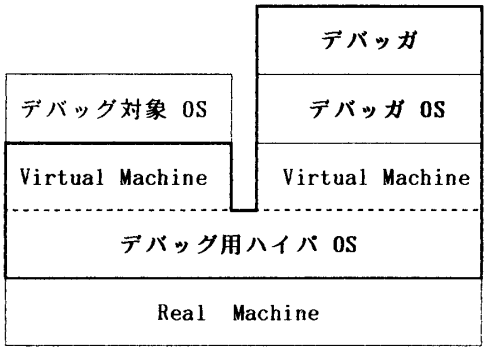

図1　OS デバッガの構成

4．1　デバッグ用ハイパ OS

デバッグ用ハイパ OS は、仮想マシン機能、多重 OS 機能をもつハイパ OS である．このハイパ OS は、I/O インターフェースを完全に仮想化し、全ての割込みを管理している．この機能を使用して、複数の OS を同時に実行できるようになっている．

また、割込み、I/O を完全に仮想化しているので、デバッグ対象 OS の詳しい情報をデバッガ OS に伝えることができる．また、実際のマシンでは起きていない、仮想的な割込みを OS に対して、起こすことができる．これらの機能を使えば、デバッグ対象 OS に特別な仕掛けを用意しなくても、デバッグ対象 OS のトレースや操作が可能になる．

デバッグ用ハイパ OS では、仮想マシン機能、多重 OS 機能の他に、次の、デバッグ用の機能を提供する．

(1)指定 OS への割込みをデバッガ OS に通知する／しない
(2)デバッガ OS から、指定 OS へ割込みをかける
(3)指定 OS の ID を得る
(4)指定 OS の仮想マシンのレジスタを操作する
(5)指定 OS の仮想マシンのプロセッサ状態を操作する
(6)指定 OS の仮想マシンを trace する／しない
(7)I/O バッファにデータを登録する
(8)指定 OS が LOCK したデバイスを強制的に UNLOCK する
(9)強制的に UNLOCK したデバイスを LOCK し直す

以上の機能を組み合わせれば、デバッグ用の複雑な機能でも用意できる．

4．2　デバッガ OS

デバッガ OS は、デバッガに対して、キーボード、ファイル等の入出力、抽象化した割込みを提供する．

このデバッガ OS における特徴を次に示す．

(1)デバッグ対象 OS に依存しない機能を提供する
(2)デバッグ用ハイパ OS の機能や割込みを、デバッグをしやすい形に抽象化して提供する

4．3　デバッガ

デバッガは、デバッガ OS のアプリケーションとして動作するプログラムであり、デバッガ OS の SVC をコールすることにより、実現される．ハイパ OS および、デバッガ OS の機能を使うことにより、2 章で挙げた機能を提供することができる．次のような機能のデバッガが考えられる．

(1)バッチ処理型のデバッガ

あらかじめ、デバッガにデバッグの手順を、記述しておくことによりデバッグやチェックを自動で行うデバッガ．

(2)対話型のデバッガ

人間が細かい操作を行いながらデバッグする、機能を提供するデバッガ．

例えば、カーネル内部でバスエラーが発生するバグを取りたいときに、従来では、細かくチェックプリントを入れる以外に方法がなかった．しかし、今回の OS デバッガでは、(1)のバッチ型のデバッガでバスエラーを見張って捕まえて、後は(2)の対話型のデバッガで、メモリダンプ、トレース等を使って、デバッグすることが可能になる．

このように、この二種のデバッガの特徴をうまく利用すれば、デバッグ効率を上げることができると考えている．

5．OS デバッガの実現

デバッグ用ハイパ OS は、当研究室の OS OS/omicron 第 2 版の実行環境であるハイパ OS 「江戸」をデバッグ用に機能拡張する形で実現する[2]．これは、日立製作所 2050/32 ワークステーション上で稼働する．

現在、ハイパ OS の設計が終了して、実現段階にある．

6．おわりに

OS デバッガによって、割込み、I/O など、かなりの OS の機能のデバッグ環境を与えることができるようになると思う．しかし、微妙なタイミングによって起こる、再現性のない動作は、同じ実マシン上で資源を分割して使っている現在のハイパ OS の構成では、完全には再現できない．しかし、割込みを完全に管理しているので、割込みの順序などの情報は得ることができる．また、OS を実働しながら、情報を追記型光ディスクにロギングするようにすれば、再現性を持たせることができ、OS の評価に利用できる．

参考文献

[1]G.R.Allred : System/370 Integrated Emulation under OS and DOS、SJCC、1971
[2]岡野裕之、他 : 多重 OS「江戸」の設計と実現、情報処理学会論文誌、Vol.30、No.8、pp.1012-1023、1989